IN THE CLAIMS

Please amend the claims as follows:

1. (Original) An integrated circuit comprising:

a multiplier to produce a product from two floating point multiplicands having a first exponent weight;

a floating point conversion unit to convert the product from the first exponent weight to a converted product with a second exponent weight;

an adder to produce a present sum from the converted product and a previous sum having the second exponent weight; and

a post-normalization unit to convert the present sum to a floating point resultant having the first exponent weight.

2. (Original) The integrated circuit of claim 1 wherein the multiplier is configured to produce a product with an exponent weight of one.

3. (Currently Amended) ~~The integrated circuit of claim 2~~ An integrated circuit comprising:

a multiplier to produce a product from two floating point multiplicands having a first exponent weight;

a floating point conversion unit to convert the product from the first exponent weight to a converted product with a second exponent weight;

an adder to produce a present sum from the converted product and a previous sum having the second exponent weight;

a post-normalization unit to convert the present sum to a floating point resultant having the first exponent weight;

wherein the multiplier is configured to produce a product with an exponent weight of one; and

wherein the floating point conversion unit is configured to convert the product from an exponent weight of one to an exponent weight of thirty-two.

4. (Currently Amended) ~~The integrated circuit of claim 1 wherein:~~ An integrated circuit comprising:

a multiplier to produce a product from two floating point multiplicands having a first exponent weight;

a floating point conversion unit to convert the product from the first exponent weight to a converted product with a second exponent weight;

an adder to produce a present sum from the converted product and a previous sum having the second exponent weight;

a post-normalization unit to convert the present sum to a floating point resultant having the first exponent weight; wherein:

the product comprises an exponent having a least significant bit weight of one and a mantissa in carry-save format; and

the adder is configured to receive a converted product having an exponent with a least significant bit weight of thirty-two and a mantissa in carry-save format.

5. (Original) The integrated circuit of claim 4 wherein the floating point conversion unit is configured to shift a mantissa of the product by a number of bit positions equal to a value of the least significant five bits of the exponent of the product.

6. (Currently Amended) ~~The integrated circuit of claim 1~~ An integrated circuit comprising:

a multiplier to produce a product from two floating point multiplicands having a first exponent weight;

a floating point conversion unit to convert the product from the first exponent weight to a converted product with a second exponent weight;

an adder to produce a present sum from the converted product and a previous sum having the second exponent weight;

a post-normalization unit to convert the present sum to a floating point resultant having the first exponent weight;

wherein the converted product comprises a three bit exponent field having a least significant bit weight of thirty-two.

7. (Original) The integrated circuit of claim 6 wherein the converted product further comprises a fifty-seven bit mantissa field in carry-save format.

8. (Original) The integrated circuit of claim 1 wherein the post-normalization unit is configured to be turned off while the adder is producing the present sum.

9. (Original) A floating point multiply-accumulate circuit comprising:

an exponent path including:

an exponent summer to sum two input exponents having a first weight to produce a product exponent;

an exponent conversion unit coupled to the output of the exponent summer, to convert the product exponent to a second weight; and

an exponent accumulation stage to choose a larger exponent from the product exponent and an accumulated exponent; and

a mantissa path including:

a mantissa multiplier to multiply two input mantissas and produce a product mantissa;

a mantissa shifter to shift the product mantissa responsive to the exponent conversion unit in the exponent path; and

a mantissa accumulator to accumulate shifted product mantissas.

10. (Currently Amended) The floating point multiply-accumulate circuit of ~~claim 9~~ claim 11 wherein the exponent conversion unit is configured to zero the least significant five bits of the product exponent.

11. (Currently Amended) ~~The floating point multiply-accumulate circuit of claim 9~~ A floating point multiply-accumulate circuit comprising:

an exponent path including:

an exponent summer to sum two input exponents having a first weight to produce a product exponent;

an exponent conversion unit coupled to the output of the exponent summer, to convert the product exponent to a second weight; and

an exponent accumulation stage to choose a larger exponent from the product exponent and an accumulated exponent; and

a mantissa path including:

a mantissa multiplier to multiply two input mantissas and produce a product mantissa;

a mantissa shifter to shift the product mantissa responsive to the exponent conversion unit in the exponent path; and

a mantissa accumulator to accumulate shifted product mantissas;

wherein the mantissa shifter is configured to shift the product mantissa by a number of bit positions equal to a value of the least significant five bits of the product exponent.

12. (Original) The floating point multiply-accumulate circuit of claim 9 wherein the mantissa accumulator comprises four-to-two compressors.

13. (Original) The floating point multiply-accumulate circuit of claim 9 further comprising a post-normalization stage to produce a normalized floating point resultant.

14. (Original) The floating point multiply-accumulate circuit of claim 13 wherein the post-normalization stage is configured to be turned off until accumulation is complete.

15. (Currently Amended) ~~The floating point multiply-accumulate circuit of claim 9~~ A floating point multiply-accumulate circuit comprising:

an exponent path including:

an exponent summer to sum two input exponents having a first weight to produce a product exponent;

an exponent conversion unit coupled to the output of the exponent summer, to convert the product exponent to a second weight; and

an exponent accumulation stage to choose a larger exponent from the product exponent and an accumulated exponent; and

a mantissa path including:

a mantissa multiplier to multiply two input mantissas and produce a product mantissa;

a mantissa shifter to shift the product mantissa responsive to the exponent conversion unit in the exponent path; and

a mantissa accumulator to accumulate shifted product mantissas;

wherein the exponent conversion unit is configured to convert the product exponent to have a least significant bit weight equal to thirty-two.

16. (Original) The floating point multiply-accumulate circuit of claim 9 wherein the product mantissa is in carry-save format.

17. (Original) The floating point multiply-accumulate circuit of claim 16 wherein the mantissa accumulator is configured to accumulate numbers in carry-save format.

18. (Original) A method of performing a multiply-accumulate operation comprising:

multiplying two floating point mantissas and summing two floating point exponents to form a product;

converting the product to have a different least significant bit weight exponent field;

accumulating the converted product; and

post-normalizing the accumulated product.

19. (Original) The method of claim 18 wherein accumulating the product comprises accumulating the product in carry-save format.

20. (Original) The method of claim 18 wherein accumulating the product comprises adding a first plurality of products with a last product, the method further comprising turning off post-normalization until the last product is accumulated.

21. (Currently Amended) ~~The method of claim 18~~ <u>A method of performing a multiply-accumulate operation comprising:</u>

<u>multiplying two floating point mantissas and summing two floating point exponents to form a product;</u>

<u>converting the product to have a different least significant bit weight exponent field;</u>

<u>accumulating the converted product; and</u>

<u>post-normalizing the accumulated product;</u>

wherein converting comprises:

shifting a mantissa of the product by an amount equal to the value of the least significant five bits of the exponent of the product; and

zeroing the least significant five bits of an exponent of the product.

22. (Currently Amended) ~~The method of claim 18~~ A method of performing a multiply-accumulate operation comprising:

multiplying two floating point mantissas and summing two floating point exponents to form a product;

converting the product to have a different least significant bit weight exponent field;

accumulating the converted product; and

post-normalizing the accumulated product;

wherein accumulating comprises:

comparing an exponent of a first converted product to an exponent of a second converted product;

conditionally shifting right by a fixed amount the mantissa of the converted product having a smaller exponent;

selecting the larger exponent as a resultant exponent; and

producing a resultant mantissa from a mantissa of the first converted product and a mantissa of the second converted product.

23. (Original) The method of claim 22 wherein conditionally shifting right comprises selecting one of two inputs of a multiplexor.

24. (Original) The method of claim 22 wherein producing a resultant mantissa comprises selecting the mantissa of the first converted product if the exponent of the first converted product is more than one greater than the exponent of the second converted product.

25. (Original) The method of claim 22 wherein producing a resultant mantissa comprises adding mantissas of the first and second converted products to produce a resultant mantissa.

26. (Original) The method of claim 22 wherein conditionally shifting right comprises:

when the exponent of the first converted product is one greater than the exponent of the second converted product, shifting a mantissa of the second converted product thirty-two bit positions to the right.

27. (New) The integrated circuit of claim 3 wherein the adder is configured to receive a converted product having an exponent with a least significant bit weight of thirty-two and a mantissa in carry-save format.

28. (New) The floating point multiply-accumulate circuit of claim 15 further comprising a post-normalization stage to produce a normalized floating point resultant.

29. (New) The method of claim 21 wherein accumulating the product comprises adding a first plurality of products with a last product, the method further comprising turning off post-normalization until the last product is accumulated.